

# Inhaltsverzeichnis

## Abkürzungsverzeichnis — XV

## Über den Autor — XVII

### 1 Einführung in AutomationML — 1

- 1.1 Von Denkmodellen zu Datenmodellen — 1
  - 1.1.1 Denkmodelle helfen uns, die Welt zu verstehen — 1
  - 1.1.2 Denkmodelle im Ingenieurskontext: ein Bild sagt mehr als tausend Worte — 3
  - 1.1.3 Aus Denkmodellen werden Datenmodelle — 3
  - 1.1.4 Grundbegriffe — 3
  - 1.1.5 Übungsaufgabe zu Denkmodellen und Datenmodellen — 7
  - 1.1.6 Die Herausforderung beim Austausch von Datenmodellen — 7
- 1.2 Etablierte Optionen des Datenaustausches — 8
  - 1.2.1 Option 1: Einigung auf zu verwendende Werkzeuge — 8
  - 1.2.2 Option 2: Paarweiser Dateiaustausch über Ex- und Importer-Schnittstellen — 9
  - 1.2.3 Option 3: Eine gemeinsame Werkzeug-Suite — 10
- 1.3 Warum AutomationML entwickelt wurde — 11
  - 1.3.1 Werkzeugzentriertes Engineering behindert den Datenabgleich — 11
  - 1.3.2 Von der Maschinenlesbarkeit zur Maschinenverständlichkeit — 14
- 1.4 Was ist AutomationML? — 15
  - 1.4.1 AutomationML ist ein Datenformat und eine Datenmodellierungssprache — 15
  - 1.4.2 AutomationML ist anwendungsunabhängig — 16
- 1.5 Ziele von AutomationML — 18
- 1.6 Innovationen von AutomationML — 20
- 1.7 Wer steht hinter AutomationML? — 23
- 1.8 Entstehungsgeschichte — 23
- 1.9 Was unterscheidet AutomationML von UML oder SysML? — 25
- 1.10 Weiterführende Literatur und AutomationML Spezifikationen — 26
  - 1.10.1 Verfügbarkeit — 26
  - 1.10.2 Die AutomationML IEC 62714 Serie — 27
  - 1.10.3 Anwendungsempfehlungen (Application Recommendations) — 28
  - 1.10.4 Whitepaper — 29
  - 1.10.5 Best practice recommendations — 29
- 1.11 Was Sie nun können sollten — 30

### 2 AutomationML – Erste Schritte — 31

- 2.1 Architektur und Modellierungsphilosophie — 31
  - 2.1.1 Einführung — 31
  - 2.1.2 AutomationML Architektur — 31
  - 2.1.3 AutomationML Subformate — 32
- 2.2 Der AutomationML Editor — 33
  - 2.2.1 Einführung und Motivation — 33

## **VIII — Inhaltsverzeichnis**

2.2.2	Übersicht über die Bereiche des AutomationML Editors —	<b>34</b>
2.2.3	Grundfunktionen des AutomationML Editors im Überblick —	<b>36</b>
2.2.4	Die erste AutomationML-Datei —	<b>36</b>
<b>2.3</b>	<b>CAEX Modellierungsphilosophie — 37</b>	
2.3.1	Objektorientierung aus Sicht des Anlagenplaners —	<b>37</b>
2.3.2	CAEX als Metamodell verstehen —	<b>38</b>
2.3.3	Fünf fundamentale Bibliotheken —	<b>39</b>
2.3.4	Instanzen – digitale Repräsentationen individueller Objekte —	<b>40</b>
2.3.5	Klassen – vordefinierte abstrakte Datenmodelle —	<b>41</b>
2.3.6	Vererbung – ein Mechanismus zur Wiederverwendung —	<b>43</b>
2.3.7	Aggregation – Grundprinzip der Komposition —	<b>46</b>
2.3.8	Kapselung – ein Ordnungsprinzip —	<b>46</b>
2.3.9	Referenzen und Relationen – Verbindungen zwischen Objekten —	<b>47</b>
2.3.10	Klassenhierarchien —	<b>48</b>
2.3.11	Modellieren von Objektsemantik – das CAEX Rollenkonzept —	<b>50</b>
2.3.12	Zusammenfassung —	<b>52</b>
<b>2.4</b>	<b>Dokumentvalidierung: Fehlersuche mit dem AML Editor — 53</b>	
2.4.1	Motivation —	<b>53</b>
2.4.2	Dokument-Validierung mit dem AutomationML-Editor —	<b>54</b>
2.4.3	Fehlerbehebung mit dem AutomationML Editor —	<b>55</b>
<b>2.5</b>	<b>Was Sie jetzt können sollten — 56</b>	
<b>2.6</b>	<b>Eine letzte Übungsaufgabe zur Vertiefung — 57</b>	

## **3 Sprachelemente von CAEX 3.0 — 59**

<b>3.1</b>	<b>Einführung — 59</b>	
<b>3.2</b>	<b>Der Bauplan von CAEX — 61</b>	
<b>3.3</b>	<b>Fünf Bibliotheken bilden die Säulen von CAEX — 62</b>	
3.3.1	Überblick —	<b>62</b>
3.3.2	AttributeTypeLib —	<b>62</b>
3.3.3	InterfaceClassLib —	<b>62</b>
3.3.4	RoleClassLib —	<b>63</b>
3.3.5	SystemUnitClassLib —	<b>63</b>
3.3.6	InstanceHierarchy —	<b>63</b>
3.3.7	Modellierungsregeln für Bibliotheken —	<b>64</b>
<b>3.4</b>	<b>Allgemeine Modellierungsprinzipien mit CAEX — 64</b>	
3.4.1	Der Unterschied zwischen CAEX und AutomationML —	<b>64</b>
3.4.2	Allgemeine AutomationML Modellierungsregeln —	<b>64</b>
3.4.3	Ein Überblick über die CAEX Sprachelemente und ihre Funktionen —	<b>65</b>
3.4.4	Identifizieren von Klassen und Instanzen —	<b>67</b>
3.4.5	Wie man ein technisches System objektorientiert analysiert, versteht und mit CAEX abbildet —	<b>68</b>
<b>3.5</b>	<b>CAEX Attribute — 70</b>	
3.5.1	Einführung —	<b>70</b>
3.5.2	Überblick über die Architektur eines Attributtyps —	<b>71</b>
3.5.3	Praxisbeispiel —	<b>72</b>

3.5.4	Schritt 1: Modellieren eines einfachen Attributtypen — 73
3.5.5	Schritt 2: Basis-Eigenschaften eines Attributtypen — 73
3.5.6	Schritt 2: Erweiterte Eigenschaften eines Attributtypen — 75
3.5.7	Schritt 3: RefSemantic – Modellieren der Semantik eines Attributes — 75
3.5.8	Schritt 4: Modellieren von Constraints — 76
3.5.9	Schritt 5: Modellieren von Vererbungsbeziehungen — 78
3.5.10	Schritt 6: Vater-Kind-Beziehungen zwischen Attributtypen — 79
3.5.11	Schritt 7: Modellieren komplexer Attributstrukturen — 80
3.5.12	Modellieren von Attributtyp-Bibliotheken — 81
3.5.13	Eine letzte Übungsaufgabe zu Attributen — 82
3.6	<b>CAEX Schnittstellen — 83</b>
3.6.1	Einführung — 83
3.6.2	Überblick über die Architektur einer Schnittstellenklasse — 84
3.6.3	Praxisbeispiel — 85
3.6.4	Schritt 1: Modellieren einer einfachen Schnittstellenklasse — 85
3.6.5	Schritt 2: Modellieren von Schnittstellenattributen — 86
3.6.6	Schritt 3: Modellieren von Vererbung — 87
3.6.7	Schritt 4: Modellieren von komplexen geschachtelten Schnittstellen — 88
3.6.8	Schritt 5: Modellieren einer Schnittstellen-Klassenhierarchie — 88
3.6.9	Schritt 6: Verbinden von Schnittstellen: die CAEX InternalLinks — 89
3.6.10	Übungsaufgaben — 89
3.6.11	Zusammenfassung der Modellierungsregeln für Schnittstellenklassen — 91
3.7	<b>CAEX Rollen — 91</b>
3.7.1	Einführung — 91
3.7.2	Überblick über die Architektur einer Rollenklasse — 92
3.7.3	Praxisbeispiel — 92
3.7.4	Schritt 1: Modellieren einer einfachen Rollenklasse — 93
3.7.5	Schritt 2: Modellieren von Attributen und Schnittstellen einer Rollenklasse — 93
3.7.6	Schritt 3: Vererbung zwischen Rollenklassen — 94
3.7.7	Schritt 4: Modellieren von Rollenklassenhierarchien — 96
3.7.8	Zusammenfassung der Modellierungsregeln für Rollenklassen — 96
3.7.9	Übungsaufgaben — 96
3.8	<b>Modellierung von SystemUnit-Klassen — 96</b>
3.8.1	Einführung — 96
3.8.2	Überblick über die Architektur einer SystemUnitClass — 97
3.8.3	Praxisbeispiel — 99
3.8.4	Schritt 1: Modellieren einer einfachen SystemUnit-Klasse — 99
3.8.5	Schritt 2: Modellieren von Attributen und Schnittstellen — 100
3.8.6	Schritt 3: Modellieren von Vererbung — 100
3.8.7	Schritt 4: Modellierung von Aggregationen — 101
3.8.8	Schritt 5: Modellieren einer SystemUnit-Klassenhierarchie — 102
3.8.9	Schritt 6: Modellieren, Editieren und löschen von Rollenreferenzen — 103
3.8.10	Zusammenfassung der Modellierungsregeln für SystemUnit-Klassen — 105
3.9	<b>Modellieren von Instanzen in einer Instanzhierarchie — 106</b>

3.9.1	Einführung — 106
3.9.2	Drei Möglichkeiten zur Modellierung einer Instanzenhierarchie — 107
3.9.3	Die Architektur eines CAEX InternalElement — 108
3.9.4	Praxisbeispiel — 109
3.9.5	Schritt 1: Modellieren eines InternalElements — 109
3.9.6	Schritt 2: Modellieren, Ändern oder Löschen einer Rollenreferenz — 112
3.9.7	Schritt 3: Modellieren von Attributen und Schnittstellen — 112
3.9.8	Schritt 4: Modellieren von Anforderungen — 112
3.9.9	Schritt 6: Modellieren von Mapping-Objekten — 114
3.9.10	Schritt 7: Zuordnung von RoleRequirements oder SupportedRoleClass zu einem InternalElement — 116
3.9.11	Zusammenfassung der Modellierungsregeln — 118
3.10	Modellieren von Pfaden — 120
3.11	Modellieren von Versionsinformationen — 121
3.11.1	Überblick — 121
3.11.2	AutomationML Edition 2 — 121
3.11.3	AutomationML Version, CAEX SchemaVersion, Filename — 122
3.11.4	SourceDocumentInformation: der Absender eines AML Dokuments — 122
3.11.5	Versionsinformationen für Bibliotheken, Klassen, Instanzen, Attribute — 125
3.11.6	SourceObjectInformation: Referenzieren von Quellobjekten — 128
3.12	Überschreiben vererbter Informationen — 130
3.12.1	Übersicht — 130
3.12.2	Überschreiben von Attributen — 131
3.12.3	Überschreiben und Ausschließen von InternalElements/ExternalInterfaces — 133
3.13	Empfehlungen zum Modellieren mit CAEX — 134
3.13.1	Zwischenfazit — 134
3.13.2	Empfehlungen für das Analysieren von Objektmodellen — 134
3.13.3	Überblick über die wichtigsten Sprachelemente von CAEX — 135
3.14	Komplexaufgaben — 135
3.14.1	Aufgabe 1: Modellierung einer Fertigungszelle — 135
3.14.2	Aufgabe 2: Modellieren eines Motoren-Produktkataloges — 139
3.15	Was Sie jetzt können sollten — 140

## **4 AutomationML Standardbibliotheken — 141**

4.1	Einleitung — 141
4.2	Die AutomationML Standardklassen im Überblick — 141
4.3	Download der Standardbibliotheken mit dem AML Editor — 143
4.4	Download von ergänzenden Inhalten zu diesem Buch — 144
4.5	Modellierungsregeln für Standardbibliotheken — 145
4.6	Rollenklassen der AutomationMLBaseRoleClassLib — 146
4.6.1	Rollenklasse AutomationMLBaseRole — 146
4.6.2	Rollenklasse Group — 146
4.6.3	Rollenklasse Facet — 146
4.6.4	Rollenklasse Resource — 147

4.6.5	Rollenklasse Product — 147
4.6.6	Rollenklasse Process — 148
4.6.7	Rollenklasse Structure — 148
4.6.8	Rollenklasse ProductStructure — 148
4.6.9	Rollenklasse ProcessStructure — 149
4.6.10	Rollenklasse ResourceStructure — 149
4.6.11	Rollenklasse ExternalData — 149
4.7	Klassen der AutomationMLInterfaceClassLib — 150
4.7.1	Schnittstellenklasse AutomationMLBaseInterface — 150
4.7.2	Schnittstellenklasse Order — 150
4.7.3	Schnittstellenklasse Port — 150
4.7.4	Schnittstellenklasse PPRConnector — 151
4.7.5	Schnittstellenklasse ExternalDataConnector — 151
4.7.6	Schnittstellenklasse COLLADAInterface — 152
4.7.7	Schnittstellenklasse PLCopenXMLInterface — 152
4.7.8	Schnittstellenklasse ExternalDataReference — 153
4.7.9	Schnittstellenklasse Communication — 153
4.7.10	Schnittstellenklasse SignalInterface — 153
4.8	Die AutomationMLBaseAttributeTypeLib — 154
4.9	Übungsaufgaben — 156
4.10	Was Sie jetzt können sollten — 156
<b>5</b>	<b>Referenzieren externer Dokumente — 157</b>
5.1	Einführung — 157
5.2	Referenzieren externer CAEX-Dokumente — 159
5.2.1	Manuelles Referenzieren einer externen CAEX-Datei mit dem AML Editor — 159
5.2.2	Manuelles Referenzieren einer Klasse aus einer externen CAEX-Bibliothek — 160
5.2.3	Splitten und Zusammenführen von CAEX-Dokumenten — 160
5.2.4	Modellierungsregeln für das Referenzieren externer CAEX-Dokumente — 161
5.2.5	Übungsaufgabe — 162
5.3	Referenzieren von Geometrie-Dateien — 163
5.3.1	Download der benötigten Bibliotheken — 163
5.3.2	Modellieren der Geometrie-Referenz — 164
5.3.3	Übungsaufgabe — 165
5.3.4	Modellierungsregeln für Geometrie-Referenzen — 166
5.4	Referenzieren von PLCopenXML Verhaltens-Beschreibungen — 167
5.4.1	Modellieren einer PLCopenXML Referenz — 167
5.4.2	Übungsaufgabe — 168
5.4.3	Modellierungsregeln für Verhaltens-Referenzen — 169
5.5	Referenzieren von weiteren Dokumenten — 169
5.5.1	Motivation — 169
5.5.2	Modellieren von Referenzen auf externe Dokumente - Übersicht — 169
5.5.3	Modellieren der Dokumentart – das Attribut MIMETYPE — 170
5.5.4	Modellieren der Dokumentsprache – das Attribut DocLang — 171

5.5.5	Übungsbeispiel — 172
5.5.6	Referenzieren mehrerer Dokumente — 173
5.5.7	Referenzieren von Attributen in externen Dokumenten — 174
5.5.8	Modellierungsregeln für externe Dokumente außerhalb des AML Standards — 176
5.6	Übungsbeispiel: Referenzieren eines Produktbildes — 177
5.6.1	Motivation — 177
5.6.2	Übungsbeispiel — 177
5.7	Was Sie jetzt können sollten — 178
<b>6</b>	<b>Erweiterte AutomationML-Konzepte — 179</b>
6.1	Übersicht — 179
6.2	Internationalisierung, mehrsprachige Attribute — 179
6.2.1	Motivation — 179
6.2.2	Modellierungsprinzip — 179
6.2.3	Modellieren mehrsprachiger Attributtypen — 181
6.2.4	Übungsaufgaben — 181
6.2.5	Zusammenfassung der Modellierungsregeln für mehrsprachige Attribute — 182
6.3	Modellieren von Listen und Arrays — 182
6.3.1	Motivation — 182
6.3.2	Modellierungsprinzip für Listen — 182
6.3.3	Modellierungsprinzip für Arrays — 183
6.3.4	Übungsaufgaben: Modellierung von Bahnpunkten — 185
6.4	Versionierung von Klassen mit dem AutomationML Editor — 186
6.4.1	Motivation — 186
6.4.2	Modellierung von Versionsbäumen mit dem AutomationML Editor — 187
6.4.3	Übungsaufgabe — 189
6.5	Das Mirror-Konzept — 189
6.5.1	Motivation — 189
6.5.2	Mirror-Objekte und Masterobjekte — 190
6.5.3	Übungsaufgabe — 191
6.5.4	Visualisieren von Mirror-Inhalten — 192
6.5.5	Spiegeln von Schnittstellen und Attributen — 193
6.5.6	Übungsaufgabe — 194
6.5.7	Übung — 195
6.5.8	Modellierungsregeln für Mirror-Objekte — 195
6.6	Das AML-Port-Konzept — 197
6.6.1	Motivation — 197
6.6.2	Anwendungsbeispiel — 197
6.6.3	Modellierungsprinzip — 198
6.6.4	Modellierung des Beispielsystems — 199
6.6.5	Übungsaufgaben: Modellieren von Ports mit dem AML Editor — 199
6.7	Das AML-Facetten-Konzept — 201
6.7.1	Motivation — 201

6.7.2	Beispielhierarchie — 201
6.7.3	Übungsaufgabe: Modellieren von Facetten mit dem AML Editor — 202
6.7.4	Modellierungsregeln für Facetten — 204
6.8	Das AutomationML-Gruppen-Konzept — 205
6.8.1	Motivation — 205
6.8.2	Beispielhierarchie — 205
6.8.3	Modellierungsprinzip — 206
6.8.4	Übung: Modellieren von Gruppen mit dem AML Editor — 206
6.8.5	Modellierungsregeln für Gruppen — 207
6.9	Kombination von Facetten und Gruppen — 208
6.9.1	Motivation — 208
6.9.2	Beispielhierarchie — 208
6.9.3	Modellierungsprinzip — 210
6.9.4	Übung — 211
6.9.5	Anwendungsbeispiele für die Kombination von Gruppen und Facetten — 212
6.10	Modellieren von Prozessen: Das PPR-Konzept — 213
6.10.1	Motivation — 213
6.10.2	Beispielproduktion — 214
6.10.3	Modellierungsprinzip — 215
6.10.4	Schrittfolge zur Modellierung des PPR-Konzeptes mit AML — 216
6.10.5	Anwendungsbereiche des PPR-Konzeptes — 220
6.10.6	Übungsaufgaben — 221
6.11	Was Sie nun können sollten — 222

## 7 Einführung in die AutomationML-Programmierung — 223

7.1	Einführung — 223
7.2	Die AutomationML Engine — 223
7.3	Eine Trockenübung vorweg — 224
7.4	Vorbereiten der Programmierumgebung — 225
7.4.1	Ein leeres C# Projekt — 225
7.4.2	Einbinden der AutomationML Engine — 226
7.4.3	Ein kleines C# Projekt zum Lernen — 228
7.5	Programmieren von AutomationML in 10 Schritten — 229
7.5.1	Schritt 1: Erzeugen eines leeren CAEX Dokumentes — 229
7.5.2	Schritt 2: Erzeugen eines neuen leeren CAEX Dokumentes — 231
7.5.3	Schritt 3: Speichern eines CAEX Dokumentes — 231
7.5.4	Schritt 4: Öffnen eines CAEX Dokumentes — 232
7.5.5	Schritt 5: Erzeugen von Bibliotheken — 233
7.5.6	Schritt 6: Programmieren von Klassen — 234
7.5.7	Schritt 7: Programmieren einer Klassenstruktur — 235
7.5.8	Schritt 8: Programmieren der Instanzhierarchie — 236
7.5.9	Schritt 9: Programmieren der SourceDocumentInformation — 237
7.5.10	Schritt 10: Validieren von AutomationML Dokumenten — 237
7.6	Übersicht von Programmierbefehlen — 239
7.6.1	Datentypen — 239

## **XIV — Inhaltsverzeichnis**

7.6.2	Grundbefehle — 240
7.6.3	Bibliotheken erzeugen — 240
7.6.4	Attributtypen erzeugen — 241
7.6.5	InterfaceClass erzeugen — 242
7.6.6	RoleClass erzeugen und verwenden — 243
7.6.7	SystemUnitClass erzeugen und verwenden — 244
7.6.8	InternalElement erzeugen und verwenden — 245
7.7	Übungsaufgabe — 246
7.8	Was Sie nun können sollten — 246

## **8 AutomationML im Praxiseinsatz — 247**

8.1	Zwischenfazit: Was Sie bis jetzt können sollten — 247
8.2	Iterativer Datenaustausch mit AutomationML — 248
8.2.1	Iterationsunterstützung: versteckt, wichtig und schwierig — 248
8.2.2	Iterationsschleife Ebene 1: Austausch von technischen Daten — 248
8.2.3	Funktionen einer iterationsfähigen Datenaustausch-Infrastruktur — 250
8.2.4	Iterationsschleife Stufe 2: Der Entwicklungszyklus eines Datenformates — 251
8.3	Neue Optionen für den Austausch von Engineering-Daten — 252
8.3.1	Übersicht — 252
8.3.2	Option 4: Datenaustausch über harmonisierte Datenmodelle — 253
8.3.3	Option 5: Datenaustausch ohne harmonisierte Datenmodell — 254
8.4	Wie nutzt man AutomationML wirtschaftlich? — 256
8.4.1	Von Daten zu Informationen als Grundlage für Wertschöpfung — 256
8.4.2	Das vier-Ebenen-Modell — 257
8.4.3	Empfehlungen für den praktischen Einsatz von AutomationML — 259
8.5	AutomationML – Anwendungen in der Industrie — 261
8.6	Abgrenzung: Was AutomationML nicht leistet — 262

## **Literaturverzeichnis — 265**

## **Stichwortverzeichnis — 269**